

***Oath/Declaration***

1. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:  
The clause regarding "willful false statements ..." required by 37 CFR 1.68 has been omitted.

It does not identify the citizenship of each inventor.

It does not identify the city and either state or foreign country of residence of each inventor. The residence information may be provided on either an application data sheet or supplemental oath or declaration.

It does not identify the mailing address of each inventor. A mailing address is an address at which an inventor customarily receives his or her mail and may be either a home or business address. The mailing address should include the ZIP Code designation. The mailing address may be provided in an application data sheet or a supplemental oath or declaration. See 37 CFR 1.63(c) and 37 CFR 1.76.

2. Applicant is now required to submit a substitute declaration or oath to correct the deficiencies set forth in this communication. The substitute oath or declaration must be filed within the THREE MONTH shortened statutory period set for reply in the "Notice of Allowability" (PTO-37). Extensions of time may NOT be obtained under the provisions of 37 CFR 1.136. Failure to timely file the substitute declaration (or oath) will result in **ABANDONMENT** of the application. The transmittal letter accompanying the declaration (or oath) should indicate the date of the "Notice of Allowance" (PTOL-85) and the application number in the upper right hand corner.

### EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Benjamin P. Tabor (Reg. No. 60,741) on 7/1/2010.

Please amend the claims as follows:

1. (Currently Amended) In a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, a method for allocating the access of threads to a user interface context, the method comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context is configured to receive input from a user, to provide output to the user, and to maintain the context settings and the context dictionary;

determining whether the user interface context is presently being accessed by a second thread;

when the user interface context is presently being accessed by ~~[[a]]~~ the second thread, denying the request to access the user interface context received from the first thread; and

when the user interface context is not presently being accessed by [[a]] the second thread, performing a ~~processes~~ process to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record by way of [[a]] the process comprising:

(i) incident to the first thread accessing an object in the user interface context, checking a most recent entry in the context record provided within the first thread, wherein the context record identifies the contexts accessed by the first thread, and wherein the most recent entry in the context record indicates the context that is presently being accessed by the first thread;

(ii) determining whether the most recent entry in the context record matches the user interface context associated with the object being presently accessed; and

(iii) when the most recent entry in the context record does not match the user interface context associated with the object being accessed, raising an exception; and

(d) temporarily assigning to the first thread the context settings and the context dictionary maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within thread settings of the first thread upon accessing the user interface context, and wherein settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

4. (Currently Amended) The method for allocating the access of threads to a user interface context of claim [[13]] 1, the method further comprising restoring the thread settings when a thread departs the user interface context.

13. (Currently Amended) One or more computer-storage media having computer-executable instructions embodied thereon that, when executed, perform a method for allocating the access of threads to a user interface context in a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, the method for allocating the access of threads to a user interface context comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context comprises one or more objects, wherein the user interface context is configured to receive input from a user, to provide output to the user,

and to maintain the context settings and the context dictionary, wherein the context dictionary includes information from a plurality of sources;

determining whether the user interface context is presently being accessed by a second thread;

when the user interface context is presently being accessed by ~~[[a]]~~ the second thread, denying the request to access the user interface context received from the first thread; and

when the user interface context is not presently being accessed by ~~[[a]]~~ the second thread, performing a ~~processes~~ process to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record by way of ~~[[a]]~~ the process comprising:

(i) incident to the first thread accessing an object in the user interface context, checking a most recent entry in the context record provided within the first thread, wherein the context record identifies the contexts accessed by the first thread, and wherein the most recent entry in

the context record indicates the context that is presently being accessed by the first thread;

(ii) determining whether the most recent entry in the context record matches the user interface context associated with the object being presently accessed; and

(iii) when the most recent entry in the context record does not match the user interface context associated with the object being accessed, raising an exception; and

(d) temporarily assigning to the first thread the context settings and the context dictionary maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within thread settings of the first thread upon accessing the user interface context, and wherein settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

16. (Previously Presented) The one or more computer-storage media of claim 13, the method for allocating the access of threads to a user interface further comprising restoring the thread settings when a thread departs the user interface context.

-- END OF AMENDMENT --

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ERIC C. WAI whose telephone number is (571)270-1012. The examiner can normally be reached on Mon-Fri, 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng - Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric C Wai/  
Examiner, Art Unit 2195

/Li B. Zhen/  
Primary Examiner, Art Unit 2194